

Learning the Probabilistic Structure of Cumulative Phenomena with Suppes-Bayes Causal Networks

Daniele Ramazzotti, *Department of Pathology, Stanford University, CA, USA*,
 Marco S. Nobile, *Department of Informatics, University of Milano-Bicocca, Milan, Italy*,
 Marco Antonioti, *Department of Informatics, University of Milano-Bicocca, Milan, Italy*,
 and Alex Graudenzi. *Department of Informatics, University of Milano-Bicocca, Milan, Italy*.

Abstract—One of the critical issues when adopting Bayesian networks (BNs) to model dependencies among random variables is to “learn” their structure, given the huge search space of possible solutions, i.e., all the possible direct acyclic graphs. This is a well-known *NP*-hard problem, which is also complicated by known pitfalls such as the issue of *I*-equivalence among different structures. In this work we restrict the investigations on BN structure learning to a specific class of networks, i.e., those representing the dynamics of phenomena characterized by the monotonic accumulation of events. Such phenomena allow to set specific structural constraints based on Suppes’ theory of probabilistic causation and, accordingly, to define constrained BNs, named Suppes-Bayes Causal Networks (SBCNs). We here investigate the structure learning of SBCNs via extensive simulations with various state-of-the-art search strategies, such as canonical local search techniques and Genetic Algorithms. Among the main results we show that Suppes’ constraints deeply simplify the learning task, by reducing the solution search space and providing a temporal ordering on the variables.

Index Terms—Bayesian Graphical Models, Structure Learning, Probabilistic Causality, Cumulative Phenomena, Cascading Failure

I. INTRODUCTION

BAYESIAN networks (BNs) are probabilistic graphical models representing the relations of *conditional dependence* among random variables, encoded in *directed acyclic graphs* (DAGs) [1]. In the last decades, BNs have been effectively applied in several different fields and disciplines, such as

(but not limited to) diagnostics and predictive analytics [1].

One of the most challenging task with BNs is that of *learning* their structure from data. Two main approaches are commonly used to tackle this problem:

- 1) *Constraint-based* techniques: mainly due to the works by Judea Pearl [2], these approaches aim at discovering the relations of conditional independence from the data, using them as constraints to learn the network;
- 2) *Score-based* techniques: in this case the problem of learning the structure of a BN is defined as an *optimization* problem (specifically, maximization) where the search space of the valid solutions (i.e., all the possible DAGs) is evaluated via a *score* based on a *likelihood function* [1].

Regardless of the approach, the main difficulty in this learning problem is the huge number of valid solutions in the search space, namely, all the possible DAGs, which makes this task a known *NP*-hard problem, even when constraining each node to have at most two parents [3], [4]. Therefore, all state-of-the-art techniques solve this task by means of meta-heuristics [1], [5], [6].

The inference is further complicated by the well-known issue of *I-equivalence*: BNs with even very different structures can encode the same set of conditional independence properties [1]. Thus, any algorithm for structural learning can converge to a

set of equivalent structures rather than to the correct one, given that the inference itself is performed by learning the statistical relations among the variables emerging from their induced distributions rather than the structure itself [1].

In this paper, we investigate the application of BNs for the characterization of a specific class of dynamical phenomena, i.e., those driven by the *monotonic accumulation of events*. In particular, the process being modeled/observed must imply that there is: (i) a *temporal ordering* among its events (i.e., the nodes in the BN), and (ii) a monotonic accumulation over time, which probabilistically entails that the occurrence of an earlier event must be *positively correlated* to the subsequent occurrence of its successors, leading to a *significant temporal pattern* [7]. An example can be found in the dynamics of *cascading failures*, that is a failure in a system of interconnected parts where the failure of a part can trigger the failure of successive parts. This phenomenon can happen in different contexts, such as power transmission, computer networking, finance and biological systems. In these scenarios, different configurations may lead to failure, but some of them are more likely than others and, hence, can be modeled probabilistically [8].

The two particular conditions mentioned above represent the basis of the notion of *probabilistic causation* by Patrick Suppes [9], [10], and allow us to define a set of *structural constraints* to the BNs to be inferred, which, accordingly, have been dubbed as *Suppes-Bayes Causal Networks* (SBCNs) in previous works [11], [7]. SBCNs have been already applied in a number of different fields, ranging from cancer progression inference [12], [13], [14] to social discrimination discovery [11].

We specifically position our work within the aforementioned optimization-based framework for BN structure learning. The goal of this paper is to investigate how structure learning is influenced by different algorithmic choices, when representing cumulative dynamical phenomena. In particular, it is known that given a *temporal ordering* on the variables (i.e., a partially ordered set among the events, *poset* in the terminology of Bayesian networks) of a BN, finding the optimal solution that

is consistent with the ordering can be accomplished in time $O(n^k)$, where n is the number of variables and k the bounded in-degree of a node [15], [16]. Thus, the search in the *space of orderings* can be performed way more efficiently than the search in the *space of structures*, as the search space is much smaller, the branching factor is lower and acyclicity checks are not necessary [6], [17].

The determination of the right ordering in complex dynamical phenomena is generally a difficult task, which often requires considerable domain knowledge. However, the representation of cumulative phenomena via SBCNs allows to soften this hurdle, as Suppes' constraints dramatically reduce the search space of valid solutions, also providing a temporal ordering on the variables. This represents a serious theoretical advancement in structure learning of BNs for the modeling of cumulative phenomena, which we investigate in this work with a series of synthetic experiments.

In particular, in this paper we quantitatively assess the performance of learning the structure of a BN when:

- the temporal ordering among variables is given / not given, i.e., when Suppes' constraints are imposed / not imposed (in the former case we deal with SBCNs);
- different heuristic search strategies are adopted, i.e., *Hill Climbing* (HC), *Tabu Search* (TS), and *Genetic Algorithms* (GA);
- different regularization terms are used (e.g., *Bayesian Information Criterion* (BIC) and *Akaike information criterion* (AIC)).

The paper is structured as follows. In Section II we first provide a background on BNs and on the task of structure learning, with details on the classical ideas that are the foundations of our framework. Then, in Section III we describe the key features of the cumulative phenomena we aim at representing, and we formally define SBCNs as a machinery for their modeling. We also introduce an approximate algorithm for the efficient inference of SBCNs. In Section IV we make use of synthetic data to simulate such cumulative phenomena and use them to compare the performance of the various search heuristics and strategies, especially by focus-

ing on the impact of Suppes’ constraints. Finally, in Section V a discussion of the results is provided.

II. BACKGROUND

In this Section we provide an introduction to Bayesian networks together with a review of some state-of-the-art methods to tackle the problem of learning their structures from a set of observation D over the variables described in the network.

A. Bayesian graphical models

A Bayesian network is a statistical graphical model that succinctly represents a *joint distribution* over n random variables and encodes it in a *direct acyclic graph* $G = (V, E)$ over the n nodes V referring to the variables and their relations E (arcs in the DAG). Given the structure of a BN, the full joint distribution of the n variables can be written as the product of the conditional distributions on each variable. In fact, an edge between pair of nodes, e.g., A and B , denotes statistical dependence, i.e., $\mathcal{P}(A \wedge B) \neq \mathcal{P}(A)\mathcal{P}(B)$, regardless of which any other variables we condition on, that is, for any other set of variables C it holds that $\mathcal{P}(A \wedge B | C) \neq \mathcal{P}(A | C)\mathcal{P}(B | C)$ [1].

In such a DAG, the set of variables connected toward any node X determines its set of “parent” nodes $\pi(X)$. We recall that, being the graph acyclic, a node cannot be both ancestor and descendant of another node, as this would obviously cause a directed cycle. Moreover, the joint distribution over all the variables can be written as $\prod_X \mathcal{P}(X | \pi(X))$, where, if a node has no incoming edges (i.e., no parents), in the product we use its marginal probability $\mathcal{P}(X)$. Thus, to compute the probability of any combination of values over the variables, only the conditional probabilities of each variable given its parents must be parameterized. However, even in the simplest case of binary variables, the number of parameters in each conditional probability table is locally of exponential size: namely, $2^{|\pi(X)|} - 1$. Thus, the total number of parameters needed to compute the full joint distribution is of size $\sum_X 2^{|\pi(X)|} - 1$, which is anyway considerably less than $2^n - 1$ for sparse networks.

A useful property of the graphical structure is that we can define, for each variable, a set of nodes called the *Markov blanket* such that, conditioned on it, this variable is independent of all the other variables in the network. It can be proven that, for any BN, the Markov blanket consists of a node’s parents, its children and the parents of the children [1].

We also point out that the usage of the symmetrical notion of conditional dependence introduces important limitations in the task of learning the structure of a BN. As a matter of fact, we note that the two edges $A \rightarrow B$ and $B \rightarrow A$ denote equivalent dependence between A and B . Hence, two graphs having a different structure can model an identical set of independence and conditional independence relations (*I-equivalence*). This yields to the notion of *Markov equivalence class* as a *partially directed acyclic graph*, in which the edges that can take either orientation are left undirected. It is also known that two BNs are Markov equivalent when they have the same *skeleton* and the same *v-structures*, the former being the set of edges, ignoring their direction (e.g., $A \rightarrow B$ and $B \rightarrow A$ constitute a unique edge in the skeleton) and the latter being all the edge structures in which a variable has at least two parents, but those do not share an edge (e.g., $A \rightarrow B \leftarrow C$) [18].

BNs have an interesting relation to canonical boolean logical operators \wedge , \vee and \oplus and formulas over variables [19], [7]. In fact these formulas, which are “deterministic” in principle, in BNs are naturally softened into *probabilistic relations* to allow some degree of uncertainty or noise. This probabilistic approach to modeling logic allows representation of qualitative relationships among variables in a way that is inherently robust to small perturbations by noise. For instance, the phrase “*in order to hear music when listening to an mp3, it is necessary and sufficient that the power is on and the headphones are plugged in*” can be represented by a probabilistic conjunctive formulation that relates power, headphones and music, in which the probability that music is audible depends only on whether power and headphones are present. On the other hand, there is a small probability that the

music will still not play (perhaps we forgot to load any songs into the device) even if both power and headphones are on, and there is small probability that we will hear music even without power or headphone (perhaps we are next to a concert and overhear that music) [19], [20].

B. Approaches to learn the structure of a BN

In the literature, there have been two initial families of methods aimed at learning the structure of a BN from data. The methods belonging to the first family aim to explicitly *capture all the conditional independence relations* encoded in the edges, and will be referred to as *constraint based approaches* (II-B1). The second family, that of *score based approaches* (II-B2), aims at the selection of a model that *maximizes the likelihood of the data* given the model. Since both approaches lead to intractability (*NP-hardness*) [3], [4], computing and verifying an exact solution is impractical. For this reason, heuristic methods like Hill Climbing [21], Tabu Search [22], Simulated Annealing [23] and Genetic Algorithms [24], [25] are generally employed. These algorithms are characterized by a polynomial complexity, although they only provide asymptotic guarantees of converging to optimal solutions.

Recently, a third class of learning algorithms that takes advantage of *specialized logical relations* (mentioned in the previous section) have been introduced (II-B3). In the rest of this section we describe in detail some of these approaches, leaving to specific readings more detailed discussions [1], [19], [20].

1) *Constraint-based approaches*: We briefly present an intuitive explanation of several common algorithms used for structure discovery by explicitly considering conditional independence relations between variables. For more detailed explanations and analyses of complexity, correctness and stability, we refer the reader to the related references [26], [27].

The basic idea behind this class of algorithms is to build a graph structure reflecting the independence relations in the observed data, thus matching as closely as possible the empirical distribution. The difficulty in this approach lies in the number

of conditional pairwise independence tests that an algorithm would have to perform to test all possible relations. This number is indeed *exponential*, requiring to condition on a power set, when testing for the conditional independence between two variables. Because of this inherent intractability, this class of algorithms requires the introduction of some *approximations*.

2) *Score-based approaches*: This approach to structural learning aims to the maximization of the likelihood of a set of observed data. Since we assume that the data are independent and identically distributed, the likelihood of the data $\mathcal{L}(\cdot)$ is simply the product of the probability of each observation. That is,

$$\mathcal{L}(D) = \prod_{d \in D} \mathcal{P}(d)$$

for a set of observations D . Since we want to infer a model \mathcal{G} that best explains the observed data, we define the likelihood of observing the data given a specific model \mathcal{G} as:

$$\mathcal{LL}(\mathcal{G}, D) = \prod_{d \in D} \mathcal{P}(d \mid \mathcal{G}).$$

However, the actual likelihood is never used in practice, as this quantity rapidly becomes very small and impossible to represent in a computer. Instead, the logarithm of the likelihood function is usually adopted for three reasons: (i) the $\log(\cdot)$ function is monotonic; (ii) log-likelihood mitigates the numerical issues caused by normal likelihood; (iii) it is easy to compute, because the logarithm of a product is equal to the sum of the logs (e.g., $\log(xy) = \log x + \log y$), and the likelihood for a Bayesian network is a product of simple terms [1].

Practically, however, there is a problem in learning the network structure by maximizing log-likelihood alone. Namely, for any arbitrary set of data, the most likely graph is always the fully connected one (i.e., all edges are present), since adding an edge can only increase the likelihood of the data. To overcome this limitation, log-likelihood is almost always supplemented with a *regularization term* that penalizes the complexity of the model. There is a plethora of regularization terms, some based on information theory and others on Bayesian

statistics (see [28] and references therein), which all serve to promote *sparsity* in the learned graph structure, though different regularization terms are better suited for particular applications [1], [20].

3) *Learning logically constrained networks*: In Section II-A, we noted that an important class of BNs captures common binary logical operators, such as \wedge , \vee , and \oplus . Although the learning algorithms mentioned above can be used to infer the structure of such networks, some algorithms employ knowledge of these logical constraints in the learning process.

A widespread approach for the learning of monotonic progression networks with a directed acyclic graph (DAG) structure and *conjunctive events* are *Conjunctive Bayesian Networks* (see CBNs, [29]). This approach was originally adopted to model cancer progression in terms of accumulation of drivers genes [13], [14], in a way closely related to the model we discuss in this work.

This model is a standard BN over Bernoulli random variables with the constraint that the probability of a node X taking the value 1 is zero if at least one of its parents has value 0. This defines a conjunctive relationship, in that all the parents of X must be 1 for X to possibly be 1. Thus, this model alone cannot represent noise, which is an essential part of any real data. In response to this, *hidden CBNs* [30] were developed by augmenting the set of variables: a correspondence to a new variable Y that represents the observed state is assigned to each CBN variable X , which captures the “true” state. Thus, each new variable Y takes the value of the corresponding variable X with a high probability, and the opposite value with a low probability, in order to model noise observations. In this model, the variables X are latent, i.e., they are not present in the observed data, and have to be inferred from the observed values for the new variables.

III. METHOD

In this Section we present the foundations of our framework and, specifically, we define the main characteristics of the Suppes-Bayes causal networks and of some heuristic strategies for the likelihood fit. Without losing in generality, from now on, we

consider a simplified formulation of the problem of learning the structure of BNs where all the variables depicted in the graph are Bernoulli random variables, i.e., their support is $(0, 1)$. All the conclusions derived in these settings can be also directly applied to the general case where the nodes in the BN describe general random variables [1].

More precisely, we consider as an input for our learning task a dataset D of n Bernoulli variables and m cross-sectional samples. We assume the value 1 to indicate that a given variable has been observed in the sample and 0 that the variable had not been observed.

A. Suppes-Bayes causal networks

In [9], Suppes introduced the notion of *prima facie causation*. A prima facie relation between a cause u and its effect v is verified when the following two conditions are true:

- (i) *temporal priority* (TP): any cause happens before its effect;
- (ii) *probability raising* (PR): the presence of the cause raises the probability of observing its effect.

Definition 1 (Probabilistic causation, [9]). *For any two events u and v , occurring respectively at times t_u and t_v , under the mild assumptions that $0 < \mathcal{P}(u), \mathcal{P}(v) < 1$, the event u is called a prima facie cause of v if it occurs before and raises the probability of v , i.e.,*

$$\begin{cases} (TP) & t_u < t_v \\ (PR) & \mathcal{P}(v | u) > \mathcal{P}(v | \bar{u}) \end{cases} \quad (1)$$

The notion of prima facie causality has known limitations in the context of the general theories of causality [10], however, this characterizations seems to appropriate to model the dynamics of phenomena driven by the monotonic accumulation of events where a temporal order among them is implied and, thus, the occurrence of an early event positively correlates to the subsequent occurrence in time of a later one. Let us now recall again systems where cascading failure may occur: some configurations of events, in a specific order, may be more likely to cause a failure than others. This condition leads to

the emergence of an observable *temporal pattern* among the events captured by Suppes' definition of causality in terms of statistical relevance, i.e., statistical dependency.

Let us now consider a graphical representation of the aforementioned dynamics in terms of a BN $G = (V, E)$. Furthermore, let us consider a given node $v_i \in V$ and let us name $\pi(v_i)$ the set of all the nodes in V pointing to (and yet temporally preceding) v_i . Then, the joint probability distribution of the $n = |V|$ variables induced by the BN can be written as:

$$\mathcal{P}(v_1, \dots, v_n) = \prod_{v_i \in V} \mathcal{P}(v_i | \pi(v_i)) \quad (2)$$

When building our model, we need to constrain the characteristics of the considered relations as depicted in the network (i.e., the arcs in the graph), in order to account for the cumulative process above mentioned, which, in turns, needs to be reflected in its induced probability distribution [7]. To this extent, we can define a class of BNs over Bernoulli random variables named *Monotonic Progression Networks* (MPNs) [7], [19], [31]. Intuitively, MPNs represent the progression of events monotonically¹ accumulating over time, where the conditions for any event to happen is described by a probabilistic version of the canonical boolean operators, i.e., conjunction (\wedge), inclusive disjunction (\vee), and exclusive disjunction (\oplus).

Moreover, as discussed in [7], [19], [31], such MPNs can model accumulative phenomena in a probabilistic fashion, i.e., they are also modeling irregularities (noise) in the data as a small probability ε of not observing later events given their predecessors.

Given these premises, in [13] the authors describe an efficient algorithm (named CAPRI, see Algorithm 1) to learn the structure of constrained Bayesian networks which account for Suppes' criteria and which later on are dubbed *Suppes-Bayes Causal Networks* (SBCNs) in [11]. SBCNs are well suited to model cumulative phenomena as they may encode irregularities in a similar way to MPNs [7].

¹The events accumulate over time and when later events occur, earlier events are observed as well.

The efficient inference schema by Algorithm 1 rely on the observation (see [6]) that a way for circumventing the intrinsic computational complexity of the task of learning the structure of a Bayesian Network is to postulate a pre-determined ordering among the nodes. Intuitively, CAPRI exploits Suppes' theory to first mine an ordering among the nodes, reducing the complexity of the problem, and then fits the network by means of likelihood maximization. In [7] it is also shown that a SBCN, learned using Algorithm 1, can also embed the notion of accumulation through time as defined in a MPN, and, specifically, conjunctive parent sets; nevertheless SBCNs can easily be generalized to represent all the canonical boolean operators (*Extended Suppes-Bayes Causal Networks*), notwithstanding an increase of the algorithmic complexity [7]. We refer the reader to [7] for further details and, following [11], we now formally define a SBCN.

Definition 2 (Suppes-Bayes Causal Network). Given an input cross-sectional dataset D of n Bernoulli variables and m samples, the Suppes-Bayes Causal Network $SBCN = (V, E)$ subsumed by D is a directed acyclic graph such that the following requirements hold:

[Suppes' constraints] for each arc $(u \rightarrow v) \in E$ involving the selective advantage relation between nodes $u, v \in V$, under the mild assumptions that $0 < \mathcal{P}(u), \mathcal{P}(v) < 1$:

$$\mathcal{P}(u) > \mathcal{P}(v) \quad \text{and} \quad \mathcal{P}(v | u) > \mathcal{P}(v | \neg u).$$

[Simplification] let E' be the set of arcs satisfying the Suppes' constraints as before; among all the subsets of E' , the set of arcs E is the one whose corresponding graph maximizes the likelihood of the data and the adopted regularization function $R(f)$:

$$E = \arg \max_{E \subseteq E', G=(V, E)} \mathcal{LL}(G, D) - R(f). \quad (3)$$

Before moving on, we once again notice that the efficient implementation of Suppes' constraints of Algorithm 1 does not, in general, guarantee to converge to the monotonic progression networks as depicted before. In fact, the probabilistic relations of the MPNs are defined on the $\pi(v_i)$, while Algorithm

1 only considers pair of nodes rather than any pair v_i and $\pi(v_i)$. To overcome this limitation, one could extend Algorithm 1 in order to learn, in addition to the network structure, also the logical relations involving any parent set, increasing the overall computational complexity. Once again, we refer the interested reader to the discussions provided in [13], [7] and, without losing in generality, for the purpose of this work, we consider the efficient implementation of Algorithm 1.

We conclude this Section by reporting the pseudocode (see Algorithm 1) of the efficient learning algorithm to infer SBCNs presented in [13], which we adopt for the assessment of the performance in the simulations of the next Section.

Algorithm 1: CAPRI

Input: a dataset D of n Bernoulli variables.
Result: a graphical model $G = (V, E)$ representing all the relations of “probabilistic causation”.

- 1 Let $\mathcal{G} \leftarrow$ a complete graph over the vertices n ;
- 2 **forall** arcs $(u, v) \in \mathcal{G}$ **do**
- 3 Compute a score $S(\cdot)$ for the nodes u and v in terms of Suppes’ criteria;
- 4 Remove the arc (u, v) if Suppes’ criteria are not met;
- 5 **end**
- 6 Let $E \leftarrow$ the subset of the remaining arcs $E' \in \mathcal{G}$, that maximize the log-likelihood of the model, computed as: $LL(G, D) - R(f)$;
- 7 **return** The resulting graphical model $G = (V, E)$.

B. Optimization and Evolutionary Computation

As reported in line 6 of Algorithm 1, the problem of the inference of SBCNs can be re-stated as an optimization problem, in which the goal is the maximization of a likelihood score. Regardless from the strategy used in the inference process, the huge size of the search space of valid solutions makes this problem very hard to solve. Moreover, as stated above, the problem of learning the structure of a

BN is *NP*-hard [4]. Because of that, state-of-the-art techniques largely rely on heuristics [1], often based on stochastic population-based global optimization algorithms. For instance, methods based on Genetic Algorithms (GAs) [5], [17] and Ant Colony Optimization [32], [33] have been proposed in the literature.

Genetic Algorithms: GAs were introduced by Holland in 1975 [34] as a global search methodology inspired by the mechanisms of natural selection. In GA, a population \mathfrak{P} of candidate solutions (named individuals) iteratively evolves, converging towards the global optimum of a given fitness function f that, in this context, corresponds to the score to be maximized.

GAs are characterized by a well-known convergence theorem named *schema theorem*, which proves that the presence of a schema (that is, a template of solutions) in the population, i.e., positively affecting the fitness value, increases exponentially generation after generation. GAs were shown to be effective for BN learning, both in the case of available and not available *a priori* knowledge about nodes’ ordering [5], [17].

The population \mathfrak{P} is composed of Q randomly created individuals, usually represented as fixed-length strings over a finite alphabet. These strings encode putative solutions of the problem under investigation; in the case of BN learning, individuals represent linearized adjacency matrices of candidate BNs with K nodes, encoded as string of binary values whose length² is $O(K^2)$.

The individuals in \mathfrak{P} undergo an iterative process whereby three genetic operators, i.e., selection, crossover and mutation, are applied in sequence to simulate the evolution process, which results in a new population of possibly improved solutions. During the selection process, individuals from \mathfrak{P} are chosen, using a fitness-dependent sampling procedure [35]. The crossover operator is then used to recombine the structures of two promising parents taken from \mathfrak{P}' into new and improved offsprings.

²Since BNs are DAGs, the representation can be reduced by not encoding the elements on the diagonal, which are always equal to zero. In such case, the strings representing the individual have length $K \times K - K$.

The crossover is generally applied with a user-defined probability \mathcal{P}_c . Finally, the mutation operator is used to introduce new genetic materials in the population allowing a further exploration of the search space. The mutation operator replaces an arbitrary symbol of the individual, with a probability \mathcal{P}_m , with a random symbol taken from the alphabet. In the case of BNs, the mutation consists in flipping a single bit of the individual with a certain probability.

When the genetic operators have been applied, the GA can proceed by replacing the whole population with the new offspring, or by identifying the best Q individuals among parents and offspring, which are then used to create the new population.

It is worth noting that in the case of ordered nodes both crossover and mutation are *closed* operators, because the resulting offsprings always encode valid DAGs. To the aim of ensuring a consistent population of individuals throughout the generations, in the case of unordered nodes the two operators are followed by a correction procedure, in which the candidate BN is analyzed to identify the presence of invalid cycles. For further information about our implementation of GAs for the inference of BNs, including the correction phase, we refer the interested reader to [17].

IV. RESULTS

We now discuss the results of a large number experiments on simulated data with the aim of assessing the performance of the state-of-the-art score-based techniques for the BN structure inference, and comparing the performance of these methods with the learning scheme defined in Algorithm 1. Our main objective is to investigate how the performance is effected by different algorithmic choices at the different steps of the learning process.

Data generation: All simulations are performed with the following generative models. We consider 6 different topological structures.

- 1) *trees*: one predecessor at most for any node, one unique root (i.e., a node with no parents);
- 2) *forests*: likewise, more than one possible root;

- 3) *conjunctive DAGs with single root*: 3 predecessors at most for each node, all the confluences are ruled by logical conjunctions, one unique root;
- 4) *conjunctive DAGs with multiple roots*: likewise, possible multiple roots;
- 5) *disjunctive DAGs with single root*: 3 predecessors at most for each node, all the confluences are ruled by logical disjunctions, one unique root;
- 6) *disjunctive DAGs with multiple roots*: likewise, possible multiple roots;

We constrain the induced distribution of each generative structure by implying a cumulative model for either conjunctions or disjunctions, i.e., any child node cannot occur if its parent set is not activated as described for the MPN in the Method Section III. For each of these configurations, we generate 100 random structures. Furthermore, we simulate a model of noise in terms of random observations (i.e., false positives and false negatives) included in the generated datasets with different rates, as described in the next Section.

These data generation configurations are chosen to reflect: (a) different structural complexities of the models in terms of number of parameters, i.e., arcs, to be learned, (b) different types of induced distributions suitable to model cumulative phenomena as defined by the MPNs (see Section III-A), i.e., conjunction (\wedge) or inclusive disjunction (\vee)³ and, (c) situations of reduced sample sizes and noisy data.

We here provide an example of data generation. Let now n be the number of nodes we want to include in the network and let us set $p_{\min} = 0.05$ and $p_{\max} = 0.95$ as the minimum and maximum probabilities of any node. A *directed acyclic graph without disconnected components* (i.e., an instance of types (3) and (5) topologies) with maximum depth $\log n$ and where each node has at most $w^* = 3$ parents is generated as shown in Algorithm 2.

³Here we stick with the efficient search scheme of Algorithm 1 and, for this reason, we do not consider exclusive disjunction (\oplus) parent sets

Algorithm 2: Data generation: single source directed acyclic graphs

Input: n , the number of nodes of the graph, $p_{\min} = 0.05$ and $p_{\max} = 0.95$ be the minimum and maximum probabilities of any node and $w^* = 3$ the maximum incoming edges per node.

Result: a randomly generated single source directed acyclic graph.

- 1 Pick an event $r \in G$ as the root of the directed acyclic graph;
- 2 Assign to each node $u \neq r$ an integer in the interval $[2, \lceil \log n \rceil]$ representing its depth in the graph (1 is reserved for r), ensuring that each level has at least one node;
- 3 **forall** nodes $u \neq r$ **do**
- 4 Let l be the level assigned to the node;
- 5 Pick $|\mathcal{P}(u)|$ uniformly over $(0, w^*]$, and accordingly define the parents of u with events selected among those at which level $l - 1$ was assigned;
- 6 **end**
- 7 Assign $\mathcal{P}(r)$, a random value in the interval $[p_{\min}, p_{\max}]$;
- 8 **forall** events $u \neq r$ **do**
- 9 Let α be a random value in the interval $[p_{\min}, p_{\max}]$;
- 10 Let $\pi(u)$ be the direct predecessor of u ;
- 11 Then assign:

$$\mathcal{P}(u) = \alpha \mathcal{P}(x \in \pi(u));$$
- 12 **end**
- 13 **return** The generated single source directed acyclic graph.

Performance assessment: In all these configurations, the performance is assessed in terms of:

- accuracy = $\frac{(TP+TN)}{(TP+TN+FP+FN)}$;
- sensitivity = $\frac{TP}{(TP+FN)}$;
- specificity = $\frac{TN}{(FP+TN)}$;

with TP and FP being the true and false positives (we mark as positive any arc that is present in the network) and TN and FN being the true and false negatives (we mark negative any arc that is

not present in the network) with respect to the generative model. All these measures are values in $[0, 1]$ with results close to 1 indicators of good performance.

Implementation: In all the following experiments, the adopted likelihood functions (i.e., the fitness evaluations) are implemented using the *bnlearn* package [36] written in the R language, while GA [34] algorithm is implemented using the *Python* language exploiting the *inspyred* [37], *networkx* [38] and *numpy* [39] packages.

Algorithm settings: We test the performance of classical search strategies, such as Hill Climbing (HC) and Tabu Search (TS), and of more sophisticated algorithms such Genetic Algorithms (GA)⁴.

For HC and TS, we generate data as described above with networks of 10 and 15 nodes (i.e., 0/1 Bernoulli random variables). We generated 10 independent datasets for each combination of the 4 sample levels (i.e., 50, 100, 150 and 200 samples) and the 9 noise levels (i.e., from 0% to 20% with step 2.5%) for a total of 4,320,000 independent datasets. The experiments were repeated either (i) including or (ii) not including the Suppes' constraints described in Algorithm 1, and independently using 5 distinct optimization scores and regularizators, namely standard (i) log-likelihood [1], (ii) AIC [40], (iii) BIC [41], (iv) BDE [42] and (v) K2 [43], leading to a final number of 86,400,000 different configurations. While a detailed description of these regularizators is beyond the scope of this paper, we critically discuss the different performances granted by each strategy for the inference of BNs.

With respect to GA we used a restricted data generation settings, using networks of 15 nodes, datasets of 100 samples and 5 noise levels (from 0% to 20% with step 5%) for a total of 3,000 independent datasets. We tested the GA either (i) with or (ii) without Suppes' constraints, using BIC

⁴Further experiments on multi-objective optimization techniques, such as *Non-dominated Sorting Genetic Algorithm* (NSGA-II), were performed, but are not shown here because of the worse overall performance, and of the higher computational cost, with respect to canonical GA.

regularization term, leading to the final total of 6,000 different configurations. Finally, the GA was launched with a population size of 32 individuals, a mutation rate of 0.01 and 100 generations.

We summarize the performance evaluation of the distinct techniques and settings in the next Subsections and in Figures 1, 2, 3 and 4.

Performance assessment

By looking at Figure 1, one can first appreciate the variation of accuracy with respect to a specific search strategy, i.e., HC with BIC, which is taken as an example of typical behavior. In brief, the overall performance worsens with respect to: (i) a larger number of nodes in the network, (ii) more complex generative structures, and (iii) smaller samples sizes / higher noise rates. Although such a trend is intuitively expected, given the larger number of parameters to be learned for more complex models, we here underline the role of statistical complications, such as the presence of *spurious correlations* [44] and the occurrence of *Simpson’s paradox* [45].

For instance, it is interesting to observe a typical decrease of the accuracy when we compare topologies with the same properties, but different number of roots (i.e., 1 root vs. multiple roots). In the former case, we expect, in fact, a lower number of arcs (i.e., dependencies) to be learned (on average) and, hence, we may attribute the decrease of the performance to the emergence of spurious correlations among independent nodes, such as the children of the different sources of the DAG. This is due to the fact that, when sample sizes are not infinite, it is very unlikely to observe perfect independence and, accordingly, the likelihood scores may lead to overfitting. The trends displayed in Figure 1 are shared by most of the analyzed search strategies.

Role of the regularization factor: By looking at Figure 2 one can first notice that the accuracy with no regularization is dramatically lower than the other cases, as a consequence of the expected overfitting (in this case we compare the performance of HC on disjunctive DAGs with multiple

roots, but the trend is maintained in the other cases). Conversely, all regularization terms ensure the inference of sparser models, by penalizing the number of retrieved arcs.

BDE regularization term seems to be the only exception (see Figure 2), leading to unintuitive behaviors: in fact, while for all the other methods the performance decreases when higher level of noise are applied, for BDE the accuracy seems to improve with higher noise rates. This result might be explained by observing that given a topological structure, structural spurious correlations may arise between a given node and any of its undirected predecessors (i.e., one of the predecessors of its direct parents): with higher error rates, and, accordingly, more random samples in the datasets, all the correlations are reduced, hence leading to a lower impact of the regularization term. Given these considerations, one can hypothesize that the overall trend of BDE is due to a scarce penalization to the likelihood fit, favoring dense networks rather than sparse ones.

Search strategies: No significant differences in the performance between the accuracy of HC, TS and GA are observed. However, one can observe a consistent improvement in sensitivity when using GA (see Figures 3 and 4). This suggests different inherent properties of the search schemes: while with HC and TB the regularization terms, rather than the search strategy, account for most of the inference performance, GAs are capable of returning denser networks with better hit rates. This is probably due to GA’s random mutations, which allow jumps into areas of the search space characterized by excellent fitness, which could not be reached by means of greedy approaches like HC.

Suppes’ structural constraints: Finally, the most important result, which can be observed across all the different experiments, is that the overall performance of all the considered search strategies is dramatically enhanced by the introduction of Suppes’ structural constraints. In particular, as one can see, e.g., in Figure 1, there is a constant improvement in the inference, up to 10%, when

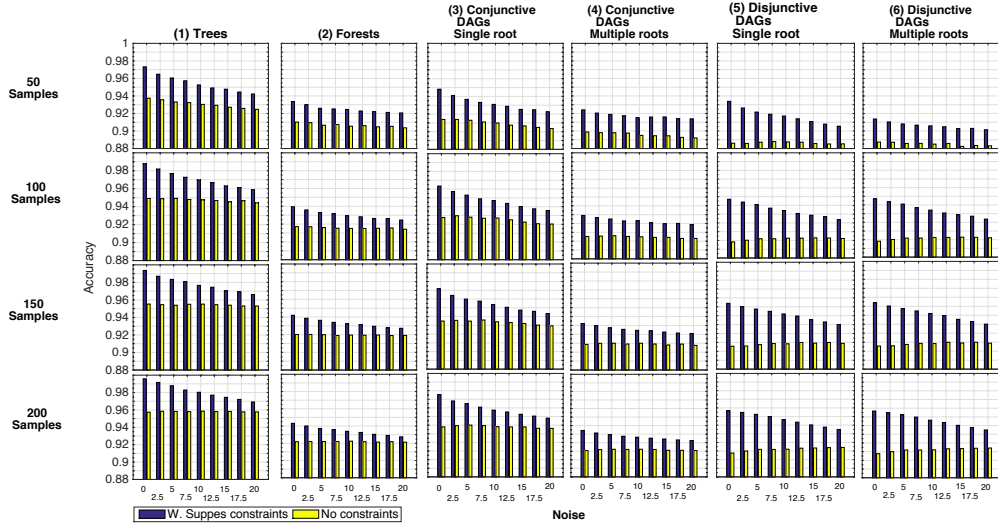


Fig. 1. Performance in terms of accuracy for the 6 considered structures with 15 nodes, noise levels from 0% to 20% with step 2.5% and sample sizes of 50, 100, 150 and 200 samples. Here we use BIC as a regularization scheme and we consider HC as a search strategy both for the classical case and when Suppes' priors are applied.

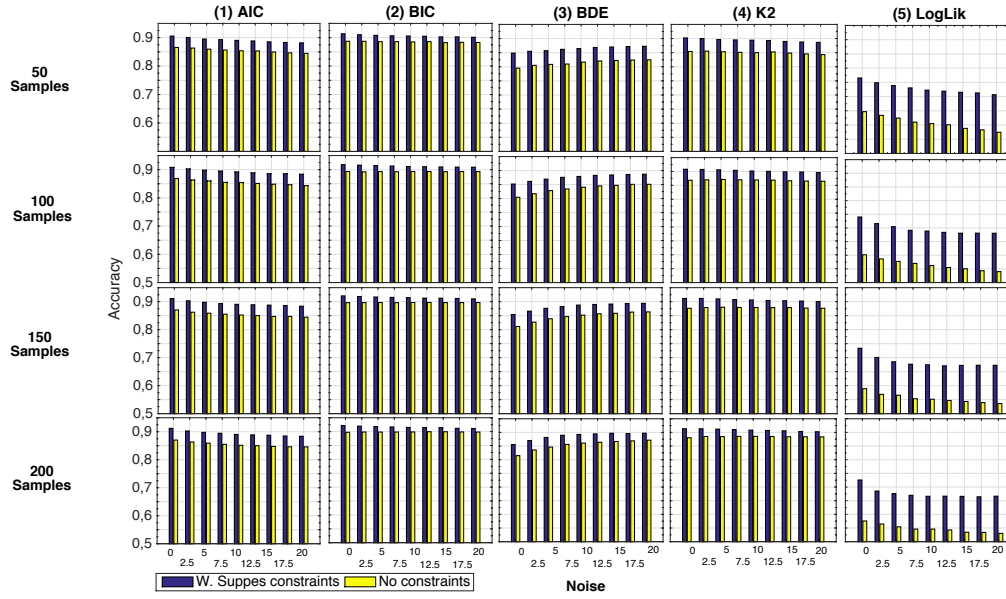


Fig. 2. Performance in terms of accuracy for directed acyclic graphs with multiple sources and disjunctive parents (structure vi) of 15 nodes, noise levels from 0% to 20% with step 2.5% and sample sizes of 50, 100, 150 and 200 samples. Here we consider HC as a search strategy both for the classical case and when Suppes' priors are applied and we show the results for all the 5 considered regularizers, i.e., the standard log-likelihood [1], AIC [40], BIC [41], BDE [42] and K2 [43].

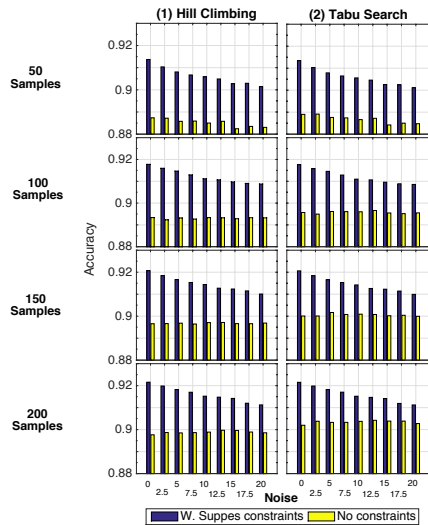


Fig. 3. Performance in terms of accuracy for directed acyclic graphs with multiple sources and disjunctive parents (structure vi) of 15 nodes, noise levels from 0% to 20% with step 2.5% and sample sizes of 50, 100, 150 and 200 samples. Here we use BIC as a regularization scheme and we consider both HC and TB as search strategies for the classical case and when Suppes’ priors are applied.

Suppes’ priors are used. Even though the accuracy of the inference is affected by the noise in the observations, in fact, the results with Suppes’ priors are consistently better than the inference with no constraints, with respect to all the considered inference settings and to all the performance measures. This is an extremely important result as it proves that the introduction of structural constraints based on Suppes’ probabilistic causation indeed simplify the optimization task, by reducing the huge search space, when dealing with BNs describing cumulative phenomena.

V. CONCLUSION

In this paper we investigated the structure learning of Bayesian Networks aimed at modeling phenomena driven by the monotonic accumulation of events over time. To this end, we made use of a subclass of constrained Bayesian networks named Suppes-Bayes Causal Networks, which in-

clude structural constraints grounded in Suppes’ theory of probabilistic causation.

While the problem of learning the structure of a Bayesian Network is known to be untractable, such constraints allow to prune the search space of the possible solutions, leading to a tremendous reduction of the number of valid networks to be considered, hence taming the complexity of the problem in a remarkable way.

We here discussed the theoretical implications of the inference process at the different steps, also by comparing various state-of-the-art algorithmic approaches and regularization methods. We finally provided an in-depth study on realistically simulated data of the effect of each inference choice, thus providing some sound guidelines for the design of efficient algorithms for the inference of models of cumulative phenomena.

According to our results, Genetic Algorithms outperform Hill Climbing and Tabu Search as search strategy, with respect to both sensitivity and specificity. Above all, we could prove that Suppes’ constraints consistently improve the inference accuracy, in all the considered scenarios and with all the inference schemes, hence positioning SBCNs as the new benchmark in the efficient inference and representation of cumulative phenomena.

REFERENCES

- [1] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.
- [2] J. Pearl, *Causality*. Cambridge university press, 2009.
- [3] D. M. Chickering, “Learning bayesian networks is np-complete,” in *Learning from data*. Springer, 1996, pp. 121–130.
- [4] D. M. Chickering, D. Heckerman, and C. Meek, “Large-sample learning of bayesian networks is np-hard,” *Journal of Machine Learning Research*, vol. 5, no. Oct, pp. 1287–1330, 2004.
- [5] P. Larranaga, M. Poza, Y. Yurramendi, R. H. Murga, and C. M. H. Kuijpers, “Structure learning of bayesian networks by genetic algorithms,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 18, no. 9, pp. 912–926, 1996.
- [6] M. Teyssier and D. Koller, “Ordering-based search: A simple and effective algorithm for learning bayesian networks,” *arXiv preprint arXiv:1207.1429*, 2012.
- [7] D. Ramazzotti, A. Graudenzi, and M. Antonietti, “Modeling cumulative biological phenomena with suppes-bayes causal networks,” *arXiv preprint arXiv:1602.07857*, 2016.

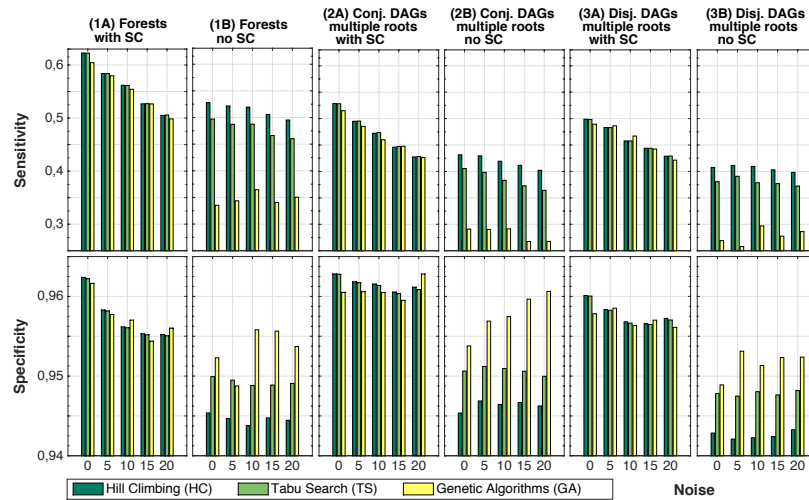


Fig. 4. Performance of HC, TB and GA, in terms of sensitivity and specificity for forests (panels 1A, 1B), directed acyclic graphs with multiple sources and conjunctive parents (panels 2A, 2B) and directed acyclic graphs with multiple sources and disjunctive parents (panels 3A, 3B) (configurations (ii), (iv) and (vi)) of 15 nodes, noise levels from 0% to 20% with step 5% and sample sizes of 100 samples. Here we use BIC as a regularization scheme for HC and TB, and for all the algorithms we either consider (panels A) or not consider (panels B) Suppes' constraints (SC).

- [8] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, and S. Havlin, "Catastrophic cascade of failures in interdependent networks," *Nature*, vol. 464, no. 7291, pp. 1025–1028, 2010.
- [9] P. Suppes, *A probabilistic theory of causality*. North-Holland Publishing Company Amsterdam, 1970.
- [10] C. Hitchcock, "Probabilistic causation," *Stanford encyclopedia of philosophy*, 2010.
- [11] F. Bonchi, S. Hajian, B. Mishra, and D. Ramazzotti, "Exposing the probabilistic causal structure of discrimination," *arXiv preprint arXiv:1510.00552*, 2015.
- [12] L. O. Loohuis, G. Caravagna, A. Graudenzi, D. Ramazzotti, G. Mauri, M. Antonioti, and B. Mishra, "Inferring tree causal models of cancer progression with probability raising," *PloS one*, vol. 9, no. 10, p. e108358, 2014.
- [13] D. Ramazzotti, G. Caravagna, L. O. Loohuis, A. Graudenzi, I. Korsunsky, G. Mauri, M. Antonioti, and B. Mishra, "Capri: efficient inference of cancer progression models from cross-sectional data," *Bioinformatics*, vol. 31, no. 18, pp. 3016–3026, 2015.
- [14] G. Caravagna, A. Graudenzi, D. Ramazzotti, R. Sanz-Pamplona, L. De Sano, G. Mauri, V. Moreno, M. Antonioti, and B. Mishra, "Algorithmic methods to infer the evolutionary trajectories in cancer progression," *PNAS*, in press, 2016.
- [15] W. Buntine, "Theory refinement on bayesian networks," in *Proceedings of the Seventh conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 1991, pp. 52–60.
- [16] G. F. Cooper and E. Herskovits, "A bayesian method for the induction of probabilistic networks from data," *Machine learning*, vol. 9, no. 4, pp. 309–347, 1992.
- [17] D. Ramazzotti, M. S. Nobile, P. Cazzaniga, G. Mauri, and M. Antonioti, "Parallel implementation of efficient search schemes for the inference of cancer progression models," in *IEEE International Conference on Computational Intelligence in Bioinformatics and Computational Biology*. IEEE, 2016.
- [18] T. V. J. Judea Pearl, "Equivalence and synthesis of causal models," in *Proceedings of Sixth Conference on Uncertainty in Artificial Intelligence*, 1991, pp. 220–227.
- [19] I. Korsunsky, D. Ramazzotti, G. Caravagna, and B. Mishra, "Inference of cancer progression models with biological noise," *arXiv preprint arXiv:1408.6032*, 2014.
- [20] D. Ramazzotti, "A model of selective advantage for the efficient inference of cancer clonal evolution," *arXiv preprint arXiv:1602.07614*, 2016.
- [21] S. S. Skiena, *The algorithm design manual: Text*. Springer Science & Business Media, 1998, vol. 1.
- [22] F. Glover, "Tabu search-part i," *ORSA Journal on computing*, vol. 1, no. 3, pp. 190–206, 1989.
- [23] S. Kirkpatrick, "Optimization by simulated annealing: Quantitative studies," *Journal of statistical physics*, vol. 34, no. 5-6, pp. 975–986, 1984.
- [24] D. E. Golberg, "Genetic algorithms in search, optimization, and machine learning," *Addison wesley*, vol. 1989, p. 102, 1989.
- [25] H. John, "Holland, adaptation in natural and artificial systems," 1992.
- [26] P. Spirtes, C. N. Glymour, and R. Scheines, *Causation, prediction, and search*. MIT press, 2000.
- [27] I. Tsamardinos, C. F. Aliferis, A. R. Statnikov, and E. Stat-

- nikov, "Algorithms for large scale markov blanket discovery," in *FLAIRS conference*, vol. 2, 2003.
- [28] A. M. Carvalho, "Scoring functions for learning bayesian networks," *Inesc-id Tec. Rep.*, 2009.
- [29] N. Beerenwinkel, N. Eriksson, and B. Sturmfels, "Conjunctive bayesian networks," *Bernoulli*, pp. 893–909, 2007.
- [30] M. Gerstung, M. Baudis, H. Moch, and N. Beerenwinkel, "Quantifying cancer progression with conjunctive bayesian networks," *Bioinformatics*, vol. 25, no. 21, pp. 2809–2815, 2009.
- [31] H. S. Farahani and J. Lagergren, "Learning oncogenetic networks by reducing to mixed integer linear programming," *PloS one*, vol. 8, no. 6, p. e65773, 2013.
- [32] L. M. De Campos, J. M. Fernandez-Luna, J. A. Gámez, and J. M. Puerta, "Ant colony optimization for learning bayesian networks," *International Journal of Approximate Reasoning*, vol. 31, no. 3, pp. 291–311, 2002.
- [33] L. M. d. Campos, J. A. Gámez Martín, and J. M. Puerta Castellón, "Learning bayesian networks by ant colony optimisation: searching in two different spaces," *Mathware & soft computing*, 2002 Vol. 9 Núm. 2 [-3], 2002.
- [34] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. U Michigan Press, 1975.
- [35] T. Back, "Selective pressure in evolutionary algorithms: A characterization of selection mechanisms," in *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*. IEEE, 1994, pp. 57–62.
- [36] M. Scutari, "Learning bayesian networks with the bnlearn r package," *arXiv preprint arXiv:0908.3817*, 2009.
- [37] A. Garrett, "inspyred: Bio-inspired algorithms in python," URL <https://pypi.python.org/pypi/inspyred>, 2012.
- [38] D. A. Schult and P. Swart, "Exploring network structure, dynamics, and function using networkx," in *Proceedings of the 7th Python in Science Conferences (SciPy 2008)*, vol. 2008, 2008, pp. 11–16.
- [39] T. Oliphant, "A guide to numpy, vol. 1," *Spanish Fork: Trelgol Publishing*, 2006.
- [40] H. Akaike, "Information theory and an extension of the maximum likelihood principle," in *Selected Papers of Hirotugu Akaike*. Springer, 1998, pp. 199–213.
- [41] G. Schwarz *et al.*, "Estimating the dimension of a model," *The annals of statistics*, vol. 6, no. 2, pp. 461–464, 1978.
- [42] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning bayesian networks: The combination of knowledge and statistical data," *Machine learning*, vol. 20, no. 3, pp. 197–243, 1995.
- [43] G. F. Cooper and E. Herskovits, "A bayesian method for constructing bayesian belief networks from databases," in *Proceedings of the Seventh conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 1991, pp. 86–94.
- [44] K. Pearson, "Mathematical contributions to the theory of evolution.—on a form of spurious correlation which may arise when indices are used in the measurement of organs," *Proceedings of the royal society of london*, vol. 60, no. 359-367, pp. 489–498, 1896.
- [45] I. Good, Y. Mittal *et al.*, "The amalgamation and geometry of two-by-two contingency tables," *The Annals of Statistics*, vol. 15, no. 2, pp. 694–711, 1987.



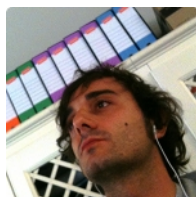
mics and Theories of Causality.



intelligence, bio-inspired algorithms, computational biology, and high-performance computing.



ence papers and the author or director of several software projects. He is the P.I. of the BIMIB group in the Dipartimento di Informatica, Sistemistica e Comunicazione dell'Università degli Studi di Milano Bicocca. Prof. Antoniotti is a member of IEEE.



the properties of biological systems and, especially, the emergence and development of cancer. He is author of numerous articles on international journals and conference proceedings.

Daniele Ramazzotti Daniele Ramazzotti received his Ph.D. in computer science from the University of Milano-Bicocca, Italy, in February 2016. He is currently a Postdoctoral Research Fellow at the Department of Pathology at the Stanford University, CA, USA. His current research interests involves Bioinformatics with specific focus on Cancer Evolution, Statistics, Bayesian Learning, Machine Learning, Algorithms and Theories of Causality.

Marco S. Nobile Marco S. Nobile received his Ph.D. in computer science from the University of Milano-Bicocca, Italy, in 2015. He is currently a Research Fellow at the Department of Informatics, Systems and Communication at the University of Milano-Bicocca, Italy. He is also member of the SYSBIO Centre of Systems Biology, Italy. His research interests include evolutionary computation, swarm intelligence, bio-inspired algorithms, computational biology, and high-performance computing.

Marco Antoniotti Prof. Marco Antoniotti obtained his M.S. and Ph.D. in Computer Science from the Courant Institute of Mathematical Sciences of New York University in 1992 and 1995. His current research interests are in Bioinformatics and Systems Biology; Simulation, Verification and Language Design Issues in Embedded and Hybrid Systems. Prof. Antoniotti is the author of several journal and conference papers and the author or director of several software projects. He is the P.I. of the BIMIB group in the Dipartimento di Informatica, Sistemistica e Comunicazione dell'Università degli Studi di Milano Bicocca. Prof. Antoniotti is a member of IEEE.

Alex Graudenzi Alex Graudenzi is an assistant professor in Computer Science at the Dept. of Informatics, Systems and Communication of the University of Milan-Bicocca. He received his Ph.D. in in Multiscale Modeling and Computational Simulation from the University of Modena and Reggio Emilia. He works at the boundaries of Informatics, Complex Systems, Statistics and Systems Biology to investigate the properties of biological systems and, especially, the emergence and development of cancer. He is author of numerous articles on international journals and conference proceedings.